# **4/1/2020 Alcor Open Source Community Meeting**

Topic: 集群横向扩展技术需求与涉及方案的讨论：控制平面的VPC力度的划分和控制平面资源的调度优化，以及容错性的一些内容

Meeting Recording: [**https://www.youtube.com/watch?v=RXLeAViXi-c&feature=youtu.be**](https://www.youtube.com/watch?v=RXLeAViXi-c&feature=youtu.be)

1. 现状及需求
   1. 现有的平台，控制集群分布在1万个节点构成的cell内，每个 controller集群管控 cell 内全部的一万节点。有多少个租户在，就需要处理多少个，VPC的规模是仅耦合。当region扩大的时候，VPC增加的话，不仅需要增加controller 集群，因为每个集群需要处理更多的北向消息，特需要增加controller数量。
2. 目标:
   1. 基于租户粒度的Vcontroller集群管控对应租户的网络配置的下发。使得集群负载压力不会随着region规模增大而提升。

## 总体方案

1. **VPC 粒度的Vcontroller集群划分:** 
   1. VPC粒度进行控制资源的划分，根据该vpc的计算资源量来配置Vcontroller的实例和管控面。提高网络配置的速度。
   2. 当租户申请了某个VPC时，为这个租户分配了Vcontroller集群。 （根据VPC的大小来分Vcontroller的集群)
   3. AZ是Active 管控面有两层，有资源管理（management port），网络配置的下放对应infrastructure pod. 都是active active的概念，不是一个备份。20 个实例的management pod是identical的
   4. 讨论/问题
      1. 根据VPC的大小来分Vcontroller，假设Vpc的资源资源量是动态？这边的假设是否一致？
         1. 扩缩容的时候调整这个平面，可以动态调整。问题不大，有控制器资源池的概念，V controller实例是资源池化。上面有个请求，（变化不是无声期）需要知道用户需要多少VM的输入信息。
         2. 创建的时候不知道VPC多大？VPC创建的时候是空白的
            1. 只要VPC的概念出来，就会有对应的Vcontroller集群。 Vcontroller规模可以调整。一开始分配的集群较小也可以扩容。
         3. 是不是所有的微服务都是这样的，都是只处理本AZ的网络资源的下放
            1. 某些微服务只处理本AZ的网络资源的下放
2. **Vcontroller 集群处理北向配置请求**
   1. 租户通过API 发送VPC的北向配置时，将配置信息放入信息队列Message queue中，然后由该vpc中的leader vcontroller实例收取该配置信息，然后由leader controller将配置任务分发给集群内的所有实例。
      1. 减少了集群处理的北向配置请求数
      2. 请求数量减少，使vcontroller集群与region 规模解耦，具备更好的横向扩展能力。
   2. Action Item:
      1. 方案设计比较各种选项。
      2. 可以并行处理的请求和不能并行处理的请求机制。
      3. Leader controller的概念是否保留，北向配置请求下来之后，每个微服务都接受这个信息，配置下发，确认思路。
   3. 讨论/问题
      1. 可以并行处理的请求和不能并行处理的请求机制。（分发/保序的机制）
      2. 跟Alcor的设计的gap，因为Alcor没有leader controller的概念，是否可以跟Alcor结合在一起做？
      3. 微服务实例都是active，identical。
      4. 沿着产品演化/还是沿着Alcor的微服务方向？
         1. 两个动态扩容的思路，如何验证还是需要更多的调研。
            1. 多个实例拆成微服务，目前是一个大集群
            2. Leader controller的概念，多个集群，每个集群一个VPC或者多个小的VPC。小集群是否还会有leader controller的概念
3. **Vcontroller 集群动态调度优化**
   1. 动态调度优化：把控制平面资源池化，可以动态调度优化。
      1. 当某些VPC的资源数量发生变化，VPC规模会随之扩容或者缩容。
         1. 缩容的时候，部分VM会被该VPC移除，关闭vcontroller集群中各实例与移除VM的映射。
         2. 扩容的时候，region内新增了部分该VPC的VM。 根据vpc规模以及vcontroller实例预估现有的vcontroller集群是否支持新增的VM，若可以，为这些VM分配已有的vcontroller实例，并构建新的映射。否则，先在AZ中为该VPC新增Vcontroller实例。
   2. Action Item:
      1. 动态扩容的思路， 需要更多的调研来验证。
         1. 微服务实例 - 多个实例拆成微服务，目前是一个大集群。可承载一个或多个VPC，共享实例（应该是无状态）
         2. 还是沿着产品演化， leader controller的概念，多个集群。每个集群一个VPC或者多个小的VPC，小集群是否还会有leader controller的概念。
   3. 讨论/问题
      1. Vcontroller的实例，是专属的还是多个VPC共享？
         1. 一个VPC对一个Vcontroller集群，
      2. 多少个VPC?
         1. 现在一个region有100万个VPC。大的少，主要是小的共享型的
4. **Vcontroller集群动态调度优化**
   1. Action Item:
      1. 重新更新一下文档 :偏向微服务设计，每个微服务有多个实例，多个实例都是active active, 可以横向扩展。

## 概要设计

1. **划分整体的流程**
   1. 更新设计方案和图。VPC创建的时候是空的，然后用户再往里面加。
   2. 管理面本身是作为一个运用情景和租户的流程不同。一堆的微服务跑在arktos的集群，然后新起一个pod 。
   3. 平台部署的API：调用计算资源的时候，可以放做在Vcontroller管理平台的一部分。
   4. 映射: 根据业务需求，如果需要，在节点上线的时候，映射需要完成。
2. **Vcontroller资源池分配实例**
   1. VPC1 新增需求700各VM，模块计算需要多少VM/CONTAINER来承载它。
   2. 有多种规格的 Vcontroller 实例可选， 每个规格有一个指导的可管控 vm 数量。
3. **低能耗的Vcontroller实例寄宿服务器的选择方案**
   1. 实例的具体创建，寄宿在哪个服务器上面。 （实例创建算法）
4. **Vcontroller 集群北向配置请求处理方案**
   1. 根据后续研发思路，做更改。
   2. 现阶段的想法：Leader controller, 监听请求，从数据库获得配置请求，来做配置下发。
5. **Vcontroller 集群的动态优化**
   1. 现阶段想法：多集群
      1. VPC规模缩容
      2. VPC规模扩容
      3. VM发生迁移
      4. 某个VPC被删除
   2. 可以更抽象的看，有一个集群，这个集群不断有请求过来。如果请求量大，就可以扩一下，如果小的话可以缩一下。应用集群有request不断过来，预估会比较困难。
   3. 为什么需要多个集群？考虑一个大的集群来run Vcontroller实例。理想的情况下是一个集群，设计做成无状态。
      1. 两个方向：
         1. 多个Vcontroller集群，有leader controller。假设存储不能share，如果是share的话也就是一套。
         2. 一个大集群，只要后端存储足够强，前端也是无状态，一个大集群没有问题，也可以scale。
            1. scale不够，需要partition。Partition 就是集群是independent。
            2. scale够的话，只要是无状态的，request挨个处理就行。
            3. 问题：

后端存储是否足够强

故障隔离

单一集群，如果布在分离很远的AZ上，数据同步会有问题。如果只布在靠的很近的AZ， 同步没有问题了，但是需要同时间管理很远的AZ的host，会有这方面的延时问题。

* 1. 每分钟有多少个VPC creation？（并不频繁） 突发的量有多大？ 理论上不会很多。VPC request不会很大，endpoint的 request会更加significant。
  2. 现在需要解决的问题是管理面的问题，而不是租户的问题：
     1. 管理面不会特别大
     2. 南向的配置下放会是一个比较大的瓶颈。如果是Vcontroller 直接和南向节点去连，每个实例的链接会很多。这边考虑可以引入一个中间件。（message queue来处理）

## 待讨论问题

1. 为了提高可用性的措施是将大型 VPC 的业务分别部署在不同的 az 中，还是部署在一个az 中，并在其余 az 中做备份。或者说它们的关系是主从还是对等的？
   1. Active active, 对等关系。
2. 若问题 1 回答是 yes，那么当 Alcor 中的主 az 中某个微服务失效时，是由候补的 az 完全替代其工作？还是只有候补 az 中对应的微服务替代其工作？
   1. 如果某个微服务失效，还是可以找到，有入口网关。
3. 大客户需求的 vpc 网络配置用一个控制器实例能否处理过来？Alcor 的 batch 操作是否需要有多个Vcontroller实例同时工作？单一VPC的Vcontroller集群中有没有必要存在多个 Vcontroller 实例？
   1. 特别大的客户，一个控制器实例是一个集群，每个微服务会有多个实例，因为是用K8S，这边可以用到一些Horizontal scaling的东西。关键的还是程序的设计，controller是否可以处理active active. 只要程序能做到两个同时工作就可以了。
4. Arktos 能否可以帮助我们从控制器资源池中创建 Vcontroller 集群实例？
   1. 控制面作为应用程序跑在微服务平台上。
   2. Arktos/ K8S, 用Yaml文件就可以起来，很吻合，不受限。
5. Alcor 的网络配置工作大多是在网络初始化的时候进行的一次性配置，之前交流时说过xdp 表项计算下发也归 Alcor 管，xdp 表项计算下发的任务量相对于网络配置应该需要的资源少很多，如果初始化配置时 Vcontroller 实例没有过载，网络运行时 Vcontroller实例大概率不会过载，那么 4.3（2）提出的动态调整有没有必要？
   1. 跟着计算资源来， 涉及到网络操作。不能假设大部分是一次性的。
6. 云服务平台的流程
   1. 现在的业务流程给租户分配一个default 容量的Vcontroller, 先是空的，然后租户再去创建VPC。 在里面不断添加子网，路由规则，添加VM。可以去尝试在云Provider的service上尝试做一些动作。
7. 客户的输入
   1. 通常的云平台操作: 创建VPC, 创建子网, 计算有个单独界面起虚机/容器, 创建虚机，放到某个特定的子网或不指定，会指定VPC ，我的VM/container 能放到哪个子网上
   2. 去看每一个API都有哪些内容，参考一下。

## 后续合作方式

1. 加强在Slack/GitHub/ 的公共平台的讨论。
   1. 实际的场景，需求以及设计的gap.
   2. 使用Alcor的 public Slack channel
   3. 另外的两个技术点和Xiaoning Ding联系，并且可以加入Arktos 和Mizar的public Slack Channel.